In the Claims

The following is a clean version of the entire set of pending claims. In accordance with 37 CFR § 1.121(c)(1)(ii), the attachment entitled "**Version with Markings to Show Changes Made**" provides marked up versions of the claims containing the newly introduced changes.

1. A computer processor capable to execute a computer instruction which locks and then unlocks a computer resource,

the computer processor being operable to lock the resource in the course of execution of the instruction before the processor has determined whether the instruction is to be executed to completion or canceled,

the processor unlocking the resource by the time the instruction processing by the processor is terminated,

the unlocking being performed whether or not the instruction is canceled.

2. The computer processor of Claim 1 wherein

the instruction execution is pipelined, and

the instruction is canceled if a trap condition occurs after the processor started processing the instruction.

---

3. (Amended) The computer processor of Claim 1 wherein:

executing the instruction comprises reading a first memory location and conditionally or unconditionally writing a second memory location; and

the resource comprises the second memory location to be written.

A1 cont.

4. (Amended) The computer processor of Claim 3 further comprising:

a cache, wherein

the cache corresponds to the resource and comprises the second memory location.

5. (Amended) The computer processor of Claim 3 wherein

the processor is operable to perform the reading before the processor has determined whether the instruction is to be canceled.

A1 end

6. The processor of Claim 1 in combination with another processor having access to the same resource.

7. (Canceled).

8. (Amended) The processor of Claim 1 wherein:

each instruction is executed in a plurality of pipeline stages, wherein the pipeline stages for each instruction include a stage ST1 in which a signal is generated by the processor to indicate whether the instruction is to be canceled due to a trap; and

when executing the instruction which locks and then unlocks the computer resource, the processor is operable to lock the computer resource before the stage ST1.

A2

9. (Amended) The processor of Claim 8 wherein

for at least some instructions including the instruction that locks and then unlocks the computer resource, ~~the~~ stage ST1 is followed by a stage ST2 in which at least one instruction result is written to an architecture storage location; and

when the processor executes ~~the instruction~~ that locks and then unlocks the computer resource, and the instruction is to be canceled, the stage ST2 is executed for the instruction to unlock the resource but writing to the architecture storage location is suppressed.

10. A computer processor comprising an interface to a cache, the interface comprising:

address and data terminals; and

one or more control terminals to lock and unlock at least a portion of the cache, the one or more control terminals being operable to indicate that the cache is not to store data but to perform an unlock operation.

11. The processor of Claim 10 in combination with said cache, the cache being connected to the address and data terminals and to the one or more control terminals.

12. The combination of Claim 11 further comprising:

a second processor having data and address terminals and one or more control terminals, wherein said terminals of the second processor are connected to the cache.

13. The combination of Claim 12 further comprising:
a memory and a circuit for caching data from the memory in the cache.

14. A method for executing a computer instruction by a computer processor, wherein the instruction locks and then unlocks a computer resource, the method comprising:
locking the resource before the processor has determined whether the instruction is to be executed to completion or canceled; and then
unlocking the resource by the time the instruction processing by the processor is terminated, wherein the unlocking is performed whether or not the instruction is canceled.

15. The method of Claim 14 wherein
the unlocking is performed after the processor has determined whether the instruction is to be canceled.

16. (Amended) The method of Claim 14 wherein
the instruction execution is pipelined, and the instruction is canceled if a trap condition occurs after the instruction processing by the processor has begun.

A3

17. (Amended) The method of Claim 14 wherein
the instruction is an atomic instruction which comprises reading a first memory location and conditionally or unconditionally writing a second memory location; and
the resource comprises the second memory location to be written.

18. (Amended) The method of Claim 17 wherein
the second memory location to be written is a cache memory location.

19. (Amended) The method of Claim 17 wherein
the reading is performed before the processor has determined whether the instruction is to be canceled.

A4

20. (New) The method of Claim 3 wherein
the first memory location and the second memory location correspond to the same
memory location.